



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/796,785	03/08/2004	Albert Gordon Smith	58083-375010 (M074)	1669
72058 7590 09/01/2009 Kilpatrick Stockton LLP- Adobe Systems, Inc. 58083 Kilpatrick Stockton LLP 1100 Peachtree Street Atlanta, GA 30309-4530				
EXAMINER ZAHRA, ASHRAF A				
ART UNIT		PAPER NUMBER		
2175				
MAIL DATE		DELIVERY MODE		
09/01/2009		PAPER		

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary

Application No.

10/796,785

Applicant(s)

SMITH ET AL.

Examiner

ASHRAF ZAHR

Art Unit

2175

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --
Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 18 June 2009.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-19 and 28-31 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-19, 28-31 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
 2. ☐ Certified copies of the priority documents have been received in Application No. _____.
 3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- 1) ☒ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) ☐ Information Disclosure Statement(s) (PTO-8508)
Paper No(s)/Mail Date _____
- 4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date _____
- 5) ☐ Notice of Informal Patent Application
- 6) ☐ Other: _____

DETAILED ACTION

1. A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on 6/18/2009 has been entered.

Response to Arguments

2. Applicant's arguments with respect to claims 1-19, 28-31 have been considered but are moot in view of the new ground(s) of rejection.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

3. Claims 1-19, 28-31 are rejected under 35 U.S.C. 103(a) as being unpatentable over Relyea, et al. US 2004/0230900 (Hereinafter, Relyea) in view of Paperny et al., US 7,559,034 (Hereinafter, Paperny).

Regarding Claim 1, Relyea discloses, “a method comprising: responsive to beginning a rich internet application (RIA) and deferring instantiation of one or more interface elements in said RIA”. Specifically, Elements are CLR objects that are generally instantiated during runtime and form a hierarchy of objects. These elements remain in the hierarchy. Some elements are instantiated only when needed (Relyea, ¶0037).

Relyea also discloses, “generating a descriptor tree having a plurality of descriptor nodes, wherein each of said plurality of descriptor nodes describes a plurality of interface elements of said RIA instantiated and visible at said beginning of said RIA”. Specifically, during tree creation, if the object referenced doesn't exist yet, the setting of that property is delayed until it does or until the tree is completely created (Relyea, ¶0064).

Relyea also discloses, “creating one or more hidden descriptor nodes in said descriptor tree describing said one or more interface elements not instantiated or visible at said beginning, wherein said hidden descriptor nodes are created, responsive to a user navigating to said one or more interface elements”. Specifically, during tree creation, if the object referenced doesn't exist yet, the setting of that property is delayed until it does or until the tree is completely created (Relyea, ¶0064).

Relyea also discloses, “and rendering said plurality of interface elements instantiated at said beginning of said RIA and said one or more interface elements not instantiated at said beginning of RIA using corresponding ones of: said plurality of descriptor nodes; and said one or more hidden descriptor nodes”. Some elements are instantiated only when needed (Relyea, ¶0037).

Paperny also discloses, "wherein , further responsive to said user navigating, instantiating said interface elements not instantiated at said beginning of said RIA". Specifically, For example, when a user clicks on a hyperlink, banner, or graphical icon, the markup language source code associated with the link may result in the instantiation of the viewer plugin-control object (Paperny, col, 11, ln 45-60). It would be obvious to one of ordinary skill in the art at the time of the invention to combine to Relyea with Paperny to allow a user to instantiate hidden objects in a descriptor tree. The motivation to do so would be because instantiation may, according to various embodiments of the present invention, be initiated in one of a number of ways including: on document load, on mouse click, or during the viewer content downloading process (Paperny, col 11, ln 28-45).

Regarding Claim 2, Relyea also discloses, "the method of claim 1 further comprising: converting said plurality of descriptor nodes into a plurality of detail objects". Specifically, Elements are CLR objects that are generally instantiated during runtime and form a hierarchy of objects. These elements remain in the hierarchy. Some elements are instantiated only when needed (Relyea, ¶0037).

Relyea also discloses, "converting said one or more hidden descriptor nodes into one or more detail objects not instantiated at said beginning of said RIA". Some elements are instantiated only when needed (Relyea, ¶0037).

Relyea also discloses, "wherein said plurality of interface elements instantiated at said beginning of said RIA and said one or more interface elements not instantiated at said beginning of said RIA are rendered directly using said plurality of detail objects and

said one or more detail not objects instantiated at said beginning of said RIA” (Relyea, ¶0037).

Regarding Claim 3, Relyea also discloses, “the method of claim 1 wherein each one of said plurality of descriptor nodes and said one or more hidden descriptor nodes contains a software method for generating each its child nodes” (Relyea, ¶0037).

Regarding Claim 4, Relyea also discloses, “the method of claim 1 further comprising: downloading executable code representing said RIA to a computer of said user responsive to said beginning of said RIA, wherein said generating and said creating use said executable code” (Relyea, ¶0028).

Regarding Claim 5, Relyea also discloses, “the method of claim 2 further comprising: storing as a plurality of stored nodes each of: said plurality of descriptor nodes; said one or more hidden descriptor nodes; said plurality of detail objects; and said one or more detail objects not instantiated at said beginning of said RIA” (Relyea, ¶0037).

Relyea also discloses, “re-rendering each of said plurality of interface elements instantiated at said beginning of said RIA and said one or more interface elements not instantiated at said beginning of said RIA from said plurality of stored nodes” (Relyea, ¶0037).

Regarding Claim 6, Relyea also discloses, “the method of claim 1 wherein said one or more hidden descriptor nodes created has a navigational relationship with a particular one of said one or more hidden interface elements not instantiated at said beginning of said RIA to which said user navigates”. (Relyea, Fig 3:304-314).

Regarding Claim 7, Relyea also discloses, “the method of claim 6 wherein said navigational relationship comprises one or more of: a direct link; an ordinal relationship; a statistical relationship; and a positional relationship” (Relyea, Fig 3:304-314).

Regarding Claim 8, Relyea also discloses, “the method of claim 1 further comprising: creating select ones of said one or more hidden descriptor nodes in said descriptor tree responsive to beginning said RIA” (Relyea, ¶0037).

Regarding Claim 9, Relyea also discloses, “a method comprising: creating a root application node of a descriptor tree, responsive to a user initiating a rich internet application (RIA) defined using procedural code and declarative code”. Specifically, Elements are CLR objects that are generally instantiated during runtime and form a hierarchy of objects. These elements remain in the hierarchy. Some elements are instantiated only when needed (Relyea, ¶0037).

Relyea also discloses, “generating a plurality of descriptor nodes for said descriptor tree, wherein each of said plurality describes an interface element currently instantiated and visible to said user on a currently visible pane of said RIA”. Specifically, Elements are CLR objects that are generally instantiated during runtime and

Art Unit: 2175

form a hierarchy of objects. These elements remain in the hierarchy. Some elements are instantiated only when needed (Relyea, ¶0037).

Relyea also discloses, “wherein each of said plurality of stacked descriptor nodes describes said interface element not instantiated and invisible to said user on said currently visible pane of said RIA and associated with said subsequent pane”.

Specifically, during tree creation, if the object referenced doesn't exist yet, the setting of that property is delayed until it does or until the tree is completely created (Relyea, ¶0064).

Relyea also discloses, “and creating a detail object from each one of: said plurality of descriptor nodes; and said plurality of stacked descriptor nodes” “and rendering said interface element using a corresponding detail object”. Specifically, Elements are CLR objects that are generally instantiated during runtime and form a hierarchy of objects. These elements remain in the hierarchy. Some elements are instantiated only when needed (Relyea, ¶0037).

Paperny also discloses, “responsive to said user navigating to a subsequent pane of said RIA, constructing a plurality of stacked descriptor nodes for said descriptor tree”. Specifically, For example, when a user clicks on a hyperlink, banner, or graphical icon, the markup language source code associated with the link may result in the instantiation of the viewer plugin-control object (Paperny, col, 11, ln 45-60). It would be obvious to one of ordinary skill in the art at the time of the invention to combine to Relyea with Paperny to allow a user to instantiate hidden objects in a descriptor tree. The motivation to do so would be because instantiation may, according to various embodiments of the present invention, be initiated in one of a number of ways including: on document load,

on mouse click, or during the viewer content downloading process (Paperny, col 11, ln 28-45).

Regarding Claim 10, Relyea also discloses, “the method of claim 9 wherein said generating comprises: generating one of said plurality of descriptor nodes for a container of said interface element not instantiated and invisible to said user on said currently visible pane of said RIA”. Specifically, Elements are CLR objects that are generally instantiated during runtime and form a hierarchy of objects. These elements remain in the hierarchy. Some elements are instantiated only when needed (Relyea, ¶0037).

Regarding Claim 11, Relyea also discloses, “the method of claim 9 wherein said association between said subsequent pane and said plurality of stacked descriptor nodes comprises one of: a direct link; an ordinal relationship; a statistical relationship; and a positional relationship” (Relyea, Fig 3:304-314).

Regarding Claim 12-19, these claims are substantially similar to claim 1-8 and are therefore rejected based upon the same reasoning used to reject claims 1-8.

Regarding Claim 28, Relyea also discloses, “a computer program product having a computer readable medium with computer program code recorded thereon, said computer program product comprising: program code for accessing executable code of a rich internet application, the executable code comprising code for instantiating a plurality of objects, each object for rendering a corresponding interface element of the rich internet application”. Specifically, Elements are CLR objects that are generally instantiated

Art Unit: 2175

during runtime and form a hierarchy of objects. These elements remain in the hierarchy. Some elements are instantiated only when needed (Relyea, ¶0037).

Relyea also discloses, “program code for identifying a subset of the plurality of objects in the executable code, the subset comprising fewer than all of the plurality of objects”. Specifically, Elements are CLR objects that are generally instantiated during runtime and form a hierarchy of objects. These elements remain in the hierarchy. Some elements are instantiated only when needed (Relyea, ¶0037).

Relyea also discloses, “program code for instantiating the objects in the subset”. Specifically, Elements are CLR objects that are generally instantiated during runtime and form a hierarchy of objects. These elements remain in the hierarchy. Some elements are instantiated only when needed (Relyea, ¶0037).

Relyea also discloses, “program code for rendering an initial view of the application using the instantiated objects”. Specifically, Elements are CLR objects that are generally instantiated during runtime and form a hierarchy of objects. These elements remain in the hierarchy. Some elements are instantiated only when needed (Relyea, ¶0037).

Paperny also discloses, “program code for instantiating at least one other object of the plurality of objects in response to user interaction with an interface element of the initial view”. “. Specifically, For example, when a user clicks on a hyperlink, banner, or graphical icon, the markup language source code associated with the link may result in the instantiation of the viewer plugin-control object (Paperny, col, 11, ln 45-60).

Paperny also discloses, “program code for rendering another view of the application using the instantiated at least one other object”. “. Specifically, For example,

Art Unit: 2175

when a user clicks on a hyperlink, banner, or graphical icon, the markup language source code associated with the link may result in the instantiation of the viewer plugin-control object (Paperny, col, 11, ln 45-60). It would be obvious to one of ordinary skill in the art at the time of the invention to combine to Relyea with Paperny to allow a user to instantiate hidden objects in a descriptor tree. The motivation do so would be because instantiation may, according to various embodiments of the present invention, be initiated in one of a number of ways including: on document load, on mouse click, or during the viewer content downloading process (Paperny, col 11, ln 28-45).

Regarding Claim 29, Relyea also discloses, “the computer program product set forth in claim 28, further comprising program code for creating a descriptor tree comprising a plurality of descriptor nodes, each node identifying an object or container of the application”. Specifically, during tree creation, if the object referenced doesn't exist yet, the setting of that property is delayed until it does or until the tree is completely created (Relyea, ¶0064).

Relyea also discloses, “wherein identifying the subset comprises (i) using the descriptor tree to identify a node whose object is for rendering an interface element that is not visible in the initial view and (ii) excluding the object from the subset of objects”. Specifically, during tree creation, if the object referenced doesn't exist yet, the setting of that property is delayed until it does or until the tree is completely created (Relyea, ¶0064).

Regarding Claim 30, Relyea also discloses, “the computer program product set forth in claim 29, wherein using the descriptor tree to identify a node whose object is for rendering an interface element that is not visible in the initial view comprises: determining if the node identifies itself as corresponding to a stacked navigation objectFor example, when a user clicks on a hyperlink, banner, or graphical icon, the markup language source code associated with the link may result in the instantiation of the viewer plugin-control object (Paperny, col, 11, ln 45-60).

Regarding Claim 31, Relyea also discloses, “the computer program product set forth in claim 29, wherein creating a descriptor tree comprising identifying a node whose object is for rendering an interface element that is not visible in the initial view as a hidden node”; and “wherein objects associated with hidden nodes are excluded from the subset”. For example, when a user clicks on a hyperlink, banner, or graphical icon, the markup language source code associated with the link may result in the instantiation of the viewer plugin-control object (Paperny, col, 11, ln 45-60).

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to ASHRAF ZAHR whose telephone number is (571)270-1973. The examiner can normally be reached on M-F 9:30 am - 6 pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner’s supervisor, William Bashore can be reached on (571)272-4088. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Art Unit: 2175

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

AAZ 8/26/09

/Ting Zhou/

Primary Examiner, Art Unit 2173